

Data alignment between buses

Patent Number: EP1026597
Publication date: 2000-08-09
Inventor(s): CROSLAND ANDREW (GB)
Applicant(s): SUN MICROSYSTEMS INC (US)
Requested Patent: JP2000267989 (JP00267989)
Application Number: EP20000300446 20000121
Priority Number(s): US19990243169 19990203
IPC Classification: G06F13/40 ; G06F13/28
EC Classification: G06F13/40D1W
Equivalents: US6330631

Abstract

A bus bridge for a computer system for bridging first and second buses includes a shift and accumulate unit. The shift and accumulate unit includes a shifter having an input connected to receive bytes from one of the first and second buses and an output providing a selectable shift to the received bytes. The shift and accumulate unit also includes an accumulator having an input connected to receive the output of the shifter and providing accumulation of selectable bits of the shifted bytes, the accumulator having an output for supplying realigned bytes to be passed to the other of the first and second buses. The combination of the shifter and the accumulator permits a desired amount of shift to be combined with the accumulation of selected bits or bytes to realign sets of bytes from one bus and to form sets of bytes for the other bus. Burst transfer is also possible by operating the shift and accumulate unit to operate in successive cycles for successive sets of input bytes from one of the buses.

Data supplied from the esp@cenet database - I2

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2000-267989
(P2000-267989A)

(43)公開日 平成12年9月29日(2000.9.29)

| | | | |
|---------------------------|-------|---------------|-----------------------|
| (51) Int.Cl. ⁷ | 識別記号 | F I | マーク ⁸ (参考) |
| G 0 6 F 13/28 | 3 1 0 | G 0 6 F 13/28 | 3 1 0 K |
| | | | 3 1 0 M |
| | 13/36 | 13/36 | 3 2 0 A |
| // G 0 6 F 12/04 | 5 4 0 | 12/04 | 5 4 0 A |

審査請求 未請求 請求項の数32 OL (全 24 頁)

(21)出願番号 特願2000-26826(P2000-26826)
(22)出願日 平成12年2月3日(2000.2.3)
(31)優先権主張番号 09/243169
(32)優先日 平成11年2月3日(1999.2.3)
(33)優先権主張国 米国(US)

(71)出願人 591064003
サン・マイクロシステムズ・インコーポレ
ーテッド
SUN MICROSYSTEMS, IN
CORPORATED
アメリカ合衆国 94303 カリフォルニア
州・パロ アルト・サン アントニオ ロ
ード・901
(74)代理人 100064621
弁理士 山川 政樹

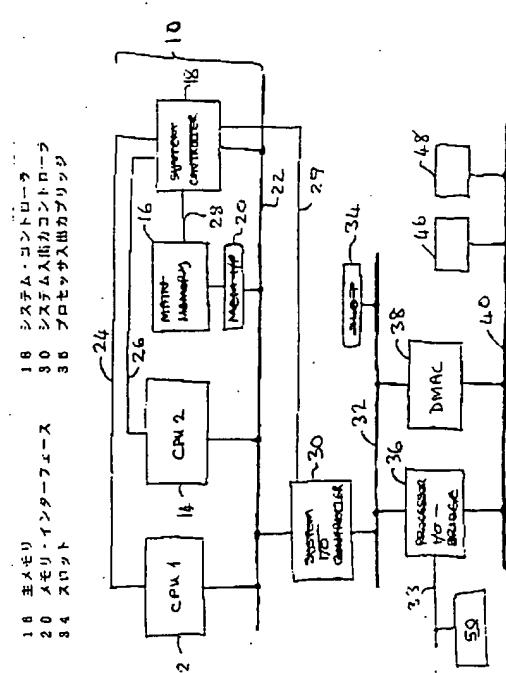
最終頁に統ぐ

(54) 【発明の名称】 バス間でのデータのアライメント

(57) 【要約】

【課題】 シフトと累積ユニットを含む、第1、第2バスをブリッジするコンピュータ・システム用のバス・ブリッジを提供する。

【解決手段】 シフトと累積ユニットは、一方のバスからバイトを受け取るために接続された入力と、受け取ったバイトを選択可能にシフトする出力とを有するシフタを含み、かつ、シフタの出力を受け取るために接続された入力と、シフトされたバイトを選択可能に累積するアキュムレータを含む。このアキュムレータが、第1、第2バスのうちの他方に渡される再アラインメントされたバイトを提供する出力を有する、シフタとアキュムレータを組み合わせることにより、所望の量のシフトと選択されたビットまたはバイトの累積とを組み合わせ、一方のバスからのバイト・セットを再アラインして、他方のバス用のバイト・セットを形成することができる。



【特許請求の範囲】

【請求項1】 第1バスと第2バスを含むコンピュータ・システム用のブリッジであって、第1バスと第2バスの間に配置されるブリッジにおいて、

第1バスおよび第2バスのうちの一方からバイトを受け取るために接続された入力と、シフトされたバイトに受け取ったバイトに関して選択可能なシフトを行う出力を有するシフタと、

シフタの出力を受け取るために接続された入力を有し、シフトされたバイトの選択的な累積を行うアキュムレータとを含み、アキュムレータが第1バスおよび第2バスのうちの他方に渡される再アラインメントされたバイトを供給する出力を有するブリッジ。

【請求項2】 シフタに接続され、受け取ったバイトに適用されるシフトを制御信号が選択する第1の制御出力と、アキュムレータに接続され、累積されるシフトされたバイトの一部を制御信号が選択する第2の制御出力を有する制御論理を含む請求項1に記載のブリッジ。

【請求項3】 制御論理が、1動作サイクル中の第1フェーズおよび第2フェーズを決める動作の連続するサイクル中にフェーズ信号を供給する出力をさらに有し、シフタが動作サイクル中に入力バイト・セットを受け取ることが可能な請求項2に記載のブリッジ。

【請求項4】 アキュムレータが出力レジスタを含み、アキュムレータが、第1フェーズで、シフトされた入力バイト・セット内の選択された位置の出力を、出力レジスタ内の対応する位置に登録することによって、再アラインメントされた出力バイト・セットの累積を完了し、第2フェーズで、シフトされた入力バイト・セット内の残りの位置からの出力を、出力レジスタ内の対応する位置に登録することによって、後続の出力バイト・セットの累積を開始するよう構成される請求項3に記載のブリッジ。

【請求項5】 1つのバイト・セットが1つのワードを形成する請求項4に記載のブリッジ。

【請求項6】 1つのバイト・セットが1つのワードの一部を形成する請求項4に記載のブリッジ。

【請求項7】 制御論理が、一方のバスから他方のバスへバイトのバースト転送を行うために、連続するサイクルおよびフェーズに対して制御信号を決定するよう構成された請求項4に記載のブリッジ。

【請求項8】 バイト・セットをバッファリングするためのバッファをさらに含む請求項1に記載のブリッジ。

【請求項9】 バイトが第1バスから第2バスへ転送される場合に、バッファが第1バスとシフタとの間に接続可能な請求項8に記載のブリッジ。

【請求項10】 バイトが第2バスから第1バスへ転送される場合に、バッファがアキュムレータと第1バスとの間に接続可能な請求項8に記載のブリッジ。

【請求項11】 第1バスからバイトを受け取るために接続された第1の入力と、アキュムレータからバイトを受け取るために接続された第2の入力と、バッファにバイトを供給するために接続された出力を有する第1のマルチプレクサを含む請求項8に記載のブリッジ。

【請求項12】 第2バスからバイトを受け取るために接続された第1の入力と、バッファの出力からバイトを受け取るために接続された第2の入力と、シフタにバイトを供給するために接続された出力を有する第2のマルチプレクサを含む請求項8に記載のブリッジ。

【請求項13】 ブリッジがダイレクト・メモリ・アクセス・コントローラを形成する請求項1に記載のブリッジ。

【請求項14】 第1バスおよび第2バスを含むコンピュータ・システムのためのブリッジであって、第1バスと第2バスの間に配置されるブリッジにおいて、第1バスおよび第2バスのうちの一方からのバイトをシフトする手段と、

第1バスおよび第2バスのうちの他方に渡される再アラインメントされたバイトを形成するためにシフト手段によってシフトされたバイトを選択的に累積する手段とを含むブリッジ。

【請求項15】 第1バスと第2バスを含むコンピュータ・システム用のダイレクト・メモリ・アクセス・コントローラであって、第1バスと第2バスとの間に配置されるダイレクト・メモリ・アクセス・コントローラにおいて、

第1バスおよび第2バスのうちの一方からバイトを受け取るために接続された入力と、受け取ったバイトに対して選択可能なシフトでシフトされたバイトを提供する出力を有するシフタと、

シフタの出力を受け取るために接続された入力を有し、シフトされたバイトを選択的に累積するアキュムレータとを含み、アキュムレータが第1バスおよび第2バスのうちの他方に渡される再アラインメントされたバイトを供給する出力を有するダイレクト・メモリ・アクセス・コントローラ。

【請求項16】 第1バス、第2バス、ならびに第1バスと第2バスとの間のブリッジを有するコンピュータ・システムであって、ブリッジが、

第1バスおよび第2バスのうちの一方からバイトを受け取るために接続された入力と、受け取ったバイトに選択可能なシフトを与える出力を有するシフタと、

シフタの出力を受け取るために接続された入力を有し、シフトされたバイトを選択的に累積するアキュムレータとを含み、アキュムレータが第1バスおよび第2バスのうちの他方に渡される再アラインメントされたバイトを供給する出力を有するコンピュータ・システム。

【請求項17】 ブリッジが、シフタに接続され、受け取ったバイトに適用されるシフトを選択する制御信号を

与える第1の制御出力と、アキュムレータに接続され、シフタの出力の一部を選択的に累積する制御信号のための第2の制御出力とを有する制御論理を含む請求項16に記載のコンピュータ・システム。

【請求項18】 制御論理が、1動作サイクル中の第1フェーズおよび第2フェーズを決めるための動作の連続するサイクル中にフェーズ信号を供給する出力をさらに有し、シフタが動作サイクル中に入力バイト・セットを受け取ることが可能な請求項17に記載のコンピュータ・システム。

【請求項19】 一方のバスから他方のバスへ、バイトのバースト転送を行うために、連続するサイクルおよびフェーズに対して制御信号を決定するように制御論理が構成されている請求項18に記載のコンピュータ・システム。

【請求項20】 アキュムレータが出力レジスタを含み、アキュムレータが、

第1フェーズで、シフタによって出力された選択可能な位置からの出力を、出力レジスタ内の対応する位置に登録することによって、再アラインメントされた出力バイト・セットの累積を完了し、

第2フェーズで、シフタの出力から、残りの位置からの出力を、出力レジスタ内の対応する位置に登録することによって、後続の再アラインメントされた出力バイト・セットの累積を開始するように構成される請求項18に記載のコンピュータ・システム。

【請求項21】 バス・ブリッジがバイト・セットをバッファリングするためのバッファをさらに含む請求項16に記載のコンピュータ・システム。

【請求項22】 バイトが第1バスから第2バスへ転送される場合に、バッファが第1バスとシフタとの間に接続可能な請求項21に記載のコンピュータ・システム。

【請求項23】 バイトが第2バスから第1バスへ転送される場合に、バッファがアキュムレータと第1バスとの間に接続可能な請求項21に記載のコンピュータ・システム。

【請求項24】 第1バスからバイトを受け取るために接続された第1の入力と、アキュムレータからバイトを受け取るために接続された第2の入力と、バッファにバイトを供給するために接続された出力とを有する第1のマルチブレクサを含む請求項21に記載のコンピュータ・システム。

【請求項25】 第2バスからバイトを受け取るために接続された第1の入力と、バッファの出力からバイトを受け取るために接続された第2の入力と、シフタにバイトを供給するために接続された出力とを有する第2のマルチブレクサを含む請求項21に記載のコンピュータ・システム。

【請求項26】 ブリッジがダイレクト・メモリ・アクセス・コントローラを形成する請求項16に記載のコン

ピュータ・システム。

【請求項27】 コンピュータ・システムにおいて第1バスと第2バスとの間でデータを転送する方法であつて、

受け取ったバイトを選択可能にシフトするように、第1バスおよび第2バスのうちの一方から受け取ったバイトをシフトするステップと、

第1バスおよび第2バスのうちの他方へ渡される再アラインメントされたバイトを供給するために、シフトされたバイトの選択的累積を実行するステップとを含む方法。

【請求項28】 連続するサイクルで使用可能であり、各サイクルで1つのバイト・セットが第1バスおよび第2バスのうちの一方から受け取られる請求項27に記載の方法。

【請求項29】 サイクルの第1フェーズで、シフトされた入力バイト・セットの選択部分を出力レジスタの対応する部分に登録することによって、再アラインメントされた出力バイト・セットの累積を完了するステップと、

出力レジスタから再アラインメントされた出力バイト・セットを出力するステップと、

サイクルの第2フェーズで、シフトされた入力バイト・セットの残りの部分を、出力レジスタの対応する部分に登録することによって、後続の再配置された出力バイト・セットの累積を開始するステップとを含む請求項28に記載の方法。

【請求項30】 第1バスから第2バスに転送する場合は、シフトステップおよび累積ステップより前にバイトがバッファされ、第2バスから第1バスに転送する場合は、シフトステップおよび累積ステップの後にバイトがバッファされる請求項27に記載の方法。

【請求項31】 第1バスと第2バスとの間でバイトのバーストを転送する方法であつて、

初期フェーズにおいて、初期出力バイト・セットを形成するように、最高2サイクルだけシフトされたバイトのうちの少なくとも選択部分が累積され、

中間フェーズにおいて、中間出力バイト・セットを形成するように、2サイクルだけシフトされたバイトのうちの少なくとも選択部分が累積され、

最終フェーズにおいて、最終出力バイト・セットを形成するように、最高2サイクルだけシフトされたバイトのうちの少なくとも選択部分が累積される請求項27に記載の方法。

【請求項32】 第1バスと第2バスとの間でバイトのバーストを転送する方法であつて、効率的なブロック転送により再アラインメントを達成するために、ブロック転送が、

選択されたビットの初期転送フェーズと、

選択されたバイトの初期転送フェーズと、

選択されたバイト・ブロックの初期転送フェーズと、
ワードのサブブロックの初期転送フェーズと、
ワードのブロック転送フェーズと、
ワードのサブブロック・ブロックの最終転送フェーズ
と、
選択されたバイト・ブロックの最終転送フェーズと、
選択されたバイトの最終転送フェーズと、
選択されたビットの最終転送フェーズとの中の少なくとも1つを含む請求項27に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・システムのバス間でのデータ転送に関する。詳細には、バス間でデータを転送するためのバス・ブリッジ、バス間でデータを転送するためのDMAコントローラ、かかるバス・ブリッジまたはDAMコントローラあるいはその両方を組み込んだコンピュータ・システム、およびバス間でデータを転送する方法に関する。

【0002】

【従来の技術】この文脈において、ブリッジとはバス間でデータを転送するための機構である。最新のコンピュータ・システムでは複数のバスが備えられていることは一般的であり、バス間でデータを転送するためにブリッジ機構が備えられていることも珍しくない。このブリッジ機構には多くの異なる形式があり、他の機能を組み込んでいる場合もある。たとえば、入出力バスと別のバスの間にDMAコントローラが備えられており、コンピュータ・システムの一部を形成していることが知られている。

【0003】

【発明が解決しようとする課題】異なるバスに対しては、異なるバイト・アドレス・アライメントを有することができる。本発明は、それぞれのバス上で選択可能なバス・アライメントを使用することによって、バス間に効率のよい転送を提供するという問題に対処する。

【0004】

【課題を解決するための手段】本発明の具体的かつ好ましい態様は、添付の特許請求の範囲の独立項および従属項で詳細に説明する。従属項の特徴の組み合わせを独立項の特徴と適切に組み合わせることが可能であり、特許請求の範囲で単に明示的に説明されているままに限らない。

【0005】本発明の一態様により、コンピュータ・システムには1つのバス・ブリッジが備えられている。コンピュータ・システムは第1のバスおよび第2のバスを含むことが可能であって、その第1のバスと第2のバスのあいだにブリッジが配置されるであろう。ブリッジは、第1のバスと第2のバスのいずれか1つからバイトを受け取るために接続される入力と、受信したバイトに

選択可能にシフト（シフト）する出力を有するシフタを含む。さらに、シフタの出力を受け取るために接続される入力を有し、シフトされたバイトを選択的に累積するアキュムレータを含む。このアキュムレータは、第1のバスおよび第2のバスの一方から他方のバスに渡すために再アラインさせたバイトを供給する出力を有する。

【0006】シフタとアキュムレータを組み合わせることにより、所望の量のシフトと選択されたバイトまたはビットの累積とを組み合わせ、一方のバスからのバイト・セットを再アラインして、他方のバス用のバイト・セットを形成することができる。この構造を使用すると、効率的な方法の任意の再アライメントによりバイトを柔軟に転送することができる。入力バスから入力された連続する入力バイト・セットに対して、シフタとアキュムレータ・ユニットを連続サイクルで動作させることにより、バースト転送も可能になる。

【0007】制御論理は、受け取ったバイトに適用されるシフトを選択するための、およびシフトされたバイトから累積されるバイトまたはビットを選択するための制御信号を提供する。シフタおよびアキュムレータは、サイクル内の連続フェーズで使用可能である。1つのサイクルでは、シフトされたバイトから、第1のバイト・セットまたはビット・セットを1つの出力バイト・セット用に選択することができる。別のサイクルでは、シフトされたバイトから、第2のバイト・セットまたはビット・セットを別の出力バイト・セット用に選択することができる。

【0008】この制御論理は、一方のバスから他方のバスへ、バイトのバースト転送を行うために、連続するサイクルやフェーズに対して制御信号を決定するように構成される。

【0009】アキュムレータは出力レジスタを含むことができる。第1のフェーズでは、シフトされた入力バイト・セットから選択した部分を、出力レジスタの対応する部分に登録することで、再アライメントされた出力バイト・セットの累積を完了させることができる。第2のフェーズでは、シフトされた入力バイト・セットの残りの部分を、出力レジスタの対応する部分に登録することで、後続の再アライメントされた出力バイト・セットの累積を開始することができる。

【0010】相対的バス幅ならびにシフトと累積ユニットの幅に応じて、1つのバイト・セットが1つのワード、またはワードの一部を形成できることに留意されたい。

【0011】本発明の実施形態では、ブリッジはバイト・セットをバッファリングするためのバッファをさらに備え、このバッファは第1バスとシフタとの間を接続することができる。このバッファは、バイトが第1バスから第2バスへ転送される場合には第1バスとシフタとの

間を接続することが可能であり、第2バスから第1バスへ転送される場合にはアキュムレータと第1バスとの間を接続することが可能である。これは、第1バスと第2バスに関するタイミングの異なる動作可能速度を考慮に入れるために、データを柔軟にバッファリングするためのものである。

【0012】前述のように、バッファ・ユニットならびにシフトおよび累積ユニットの選択順序を与えるために、バッファおよびシフタは入力マルチブレクサに関連付けられることが好ましい。

【0013】バッファ・ユニットの入力マルチブレクサは、第1バスからのバイトを受け取るために接続可能な第1の入力、アキュムレータからのバイトを受け取るために接続される第2の入力、ならびにバッファ・ユニットのバッファへバイトを出力するために接続される出力を含むことができる。

【0014】シフタの入力マルチブレクサは、第2バスからのバイトを受け取るために接続される第1の入力、バッファ・ユニットの出力からのバイトを受け取るために接続される第2の入力、ならびにシフタへバイトを出力するために接続される出力をと備えることができる。

【0015】バス・ブリッジの一実施形態では、たとえば入出力バスと別のプロセッサ・バスとの間を接続するためであり、たとえば適切なシステム・バス(SBus)プロトコルの下で使用可能なバスである、ダイレクト・メモリ・アクセス・コントローラを形成する。

【0016】本発明の他の態様では、第1バスおよび第2バスを有するコンピュータ・システム用の、ダイレクト・メモリ・アクセス・コントローラが用意されている。ダイレクト・メモリ・アクセス・コントローラは、第1バスと第2バスとの間に配置することができる。これは、第1バスおよび第2バスのうちの1つからバイトを受け取るために接続される入力を有するシフタと、受け取ったバイトに選択可能にシフトさせる出力を備える。さらに、シフタの出力を受け取るために接続される入力を有し、シフトされたバイトの選択的累積を行うアキュムレータも備え、このアキュムレータは、第1バスおよび第2バスのうちの他方へ渡すために再アラインメントされたバイトを供給するための出力を有する。

【0017】本発明の他の態様により、第1バス、第2バス、ならびに前述のような第1バスと第2バスとの間にあらバス・ブリッジを有するコンピュータ・システムが提供されている。

【0018】本発明の他の態様により、コンピュータ・システムの第1バスと第2バスとの間でデータを転送する方法が提供されている。この方法は、受け取ったバイトに対して選択可能なシフトを行うために、第1バスおよび第2バスのうちの1つから受け取ったバイトをシフトするステップと、第1バスおよび第2バスのうちの他方へ渡すために再アラインメントされたバイトを供給す

るための、シフトされたバイトの選択的累積を実行するステップとを含む。

【0019】これらのステップは連続サイクルで繰り返され、各サイクルで1つのバイト・セットが受け取られる、この1つのバイト・セットは、前述のようにワードまたはワードの一部である可能性がある。

【0020】サイクルの第1フェーズでは、シフトされた入力バイト・セットの選択位置から出力レジスタの対応する位置にバイトまたはビットを登録することによって、再アラインメントされた出力バイト・セットの累積を完了することが可能であって、サイクルの第2のフェーズでは、シフトされた入力バイト・セットの残りの位置から出力レジスタの対応する位置にバイトまたはビットを登録することで、後続の再アラインメントされた出力バイト・セットの累積を開始することが可能である。

【0021】第1バスから第2バスへ転送する場合、バイトはシフタおよびアキュムレータによって処理される前にバッファリングされ、第2バスから第1バスへ転送する場合、バイトはシフタおよびアキュムレータによって処理された後にバッファリングされる。

【0022】第1バスと第2バスの間でバイトのペーストを転送する場合、その方法は初期フェーズにおいて、初期出力バイト・セットを形成するために、最高2サイクルからのシフトされたバイトの選択ビットが累積されるステップと、中間フェーズにおいて、中間出力バイト・セットを形成するために、2サイクルからのシフトされたバイトの選択ビットが累積されるステップと、最終フェーズにおいて、最終出力バイト・セットを形成するために、最高2サイクルまでのシフトされたバイトの選択ビットを累積するステップとの中の1つまたは複数のステップを含む。

【0023】第1バスと第2バスとの間でバイトのペーストを転送する場合、ここで効率的なブロック転送によって再アラインメントを達成するために、ブロック転送は、選択されたビットの初期転送フェーズと、選択されたバイトの初期転送フェーズと、選択されたバイト・ブロックの初期転送フェーズと、ワードのサブブロックの初期転送フェーズと、ワードのブロック転送フェーズと、ワードのサブブロック・ブロックの最終転送フェーズと、選択されたバイト・ブロックの最終転送フェーズと、選択されたバイトの最終転送フェーズと、選択されたビットの最終転送フェーズとの中の、1つまたは複数を含むことができる。

【0024】

【発明の実施の形態】本発明の例示的な実施の形態を、添付の図面を参照し例示的なものとしてのみ以下に示すが、ここで参照符号などは要素などに関するものとする。

【0025】本発明の実施の形態について、フォールト・トレラント・コンピュータ・システムの一部を形成す

るのに適したコンピュータ・システムの構成例を参照しながら下記に記載する。ただし、本発明はこのような実施の形態に限定されるものではなく、DMAコントローラを採用したどのようなコンピュータ・システムにでも実装可能であることに留意されたい。

【0026】図1は、コンピュータ・システム10の例を示す概略図である。図1に示されたコンピュータ・システム10は、本実施形態ではCPU1およびCPU2である第1プロセッサ12および第2プロセッサ14を含む。代替の実施形態では、たとえば一つのCPUだけであってもよい。プロセッサ12とプロセッサ14は、UPA(Universal Processor Architecture)プロトコルの下で使用可能な第1バス22に接続されている。さらに主メモリ16も、メモリ・インターフェース20を介してこの第1バス22に接続されている。システム入出力コントローラ30は、第1バス22とシステム・バス(SBus)32との間でブリッジの役目を果たす。システム・コントローラ18は第1バス22に接続され、さらに制御ライン24、26、28、および29を介してプロセッサ12、プロセッサ14、主メモリ16、およびシステム入出力コントローラ30にもそれぞれ接続されている。

【0027】SBus32に別の構成要素(図示せず)を接続するために、1つまたは複数のスロット(たとえばスロット34)を備えることができる。プロセッサ入出力バス・ブリッジ36がSBus32に接続される。これにより、1つまたは複数の周辺装置50を接続するための周辺バス(PBus)33へ接続される。プロセッサ入出力ブリッジ36は、入出力バス40を介した1つまたは複数の周辺装置46および48へのプロセッサ入出力のアクセスを行う。

【0028】入出力バス40上にある周辺装置がプロセッサ10へ、具体的にはプロセッサ10の主メモリ16へDMAアクセスを行うために、入出力バス40とSBus32との間にダイレクト・メモリ・アクセス・コントローラ(DMAC)38も接続されている。このDMAC38は別のバス・ブリッジを形成する。

【0029】フォールト・トレラント・コンピュータ・システムの場合、入出力バスを1対のフォールト・トレラント冗長バスとして構成することができる。その際、コンピュータ・システム10の多様なインスタンスをそのバスに接続する。

【0030】図2は、DMAC38をさらに詳細に例示した図である。DMAC38は、ローダ・マシン62(以下ローダ62と呼ぶ)およびムーバ・マシン64(以下ムーバ64と呼ぶ)を備える。ローダ62とムーバ64はそれぞれ、プロセッサ入出力インターフェース(それぞれ66および68)を有し、これによってプロセッサ10がローダおよびムーバを直接制御できる。

【0031】本発明の一実施形態では、ローダ62はロ

ーダ・コントローラ84を備え、これによってローダ62が主メモリ16からDMAC38へDMAコマンドをロードできる。具体的に言えばローダ・コントローラ84は、プロセッサ10の主メモリ16からムーバ64内のムーバ・レジスタ74へDMAコマンドをロードするように動作する。

【0032】ローダ62はレジスタ72も備える。具体的にこれらは、主メモリ16内にあるコマンド・バッファ92をアドレス指定するためのベース・ポインタ75用のベース・レジスタ76と、コマンド・バッファ92内にあるDMAコマンド・シーケンスのテールをアドレス指定するテール・ポインタ77用のテール・ポインタ・レジスタ78と、コマンド・バッファ92内にあるコマンド・シーケンスのヘッドをアドレス指定するヘッド・ポインタ79用のヘッド・ポインタ・レジスタ80とを含む。ローダ62にはその他のレジスタ82も備えられている。

【0033】ムーバ64は、バス間で大量のデータを移動させるのに使用できる。これを実行するためのムーバ転送機構85を含む。ムーバ64はムーバ・コントローラ86を備える。これはムーバ・レジスタ74内のDMAコマンドに従ってDMA操作を制御するために使用可能である。ムーバ64用のDMAコマンドは、プロセッサ10によってセットアップされた16バイト構造である。DMAコマンドはムーバ64に、入出力バス40とSBusDVMA(ダイレクト仮想メモリ・アクセス)アドレスとの間でどちらかの方向にデータを移動させ、主メモリに中継する。このDMAコマンドは、プロセッサ10によってムーバ・レジスタ74に直接書き込むか、あるいは主メモリ16内のコマンド・バッファに書き込んでから、ローダ62によってムーバ・レジスタ74にロードすることができる。

【0034】DMAコマンドは、DVMAおよびフォールト・トレラント入出力バス・アクセスに任意の開始アドレスを指定することができる。これは、アライメントの制約なしに、所与の最大値までの任意の転送長さを指定することができる。DMAコマンドは、

- 主メモリのDVMAアドレスと、
- 入出力バス・アドレスと、
- バイト・カウントと、
- DVMAアクション・バイトとを含むことができる。

【0035】DVMAアクション・バイトは、(転送方向に関する)方向インジケータ、妥当性ビット、割込みイネーブル・ビット、およびライトバック・イネーブル・ビットを含むコマンドおよび状況情報を備える。

【0036】本発明の実施形態は、プロセッサ10とDMAC38との間でDMAコマンドを転送する際のオーバヘッドを最小限に抑えるために、DMAコマンドを効率的に転送するための特定の機構を含む。

【0037】図3に示すように、主メモリの領域はコマンド・バッファ92として構成される。コマンド・バッファ92として構成された主メモリ16のこの領域は、オペレーティング・システム90の制約およびDMAコントローラによる主メモリ16の使い方により、主メモリ内に配置される。たとえば、現在の例では、このコマンド・バッファは8Kバイト境界で配置された8Kバイト・ページとして定義される。図3は、DMAC 36内に保持されているベース・レジスタ76、テール・レジスタ78、およびヘッド・レジスタ80も例示している。プロセッサは、ミラー・ベース・レジスタ106、ミラー・ヘッド・レジスタ110、およびミラー・テール・レジスタ108内にそれぞれミラー・ベース・ポインタ105、ミラー・ヘッド・ポインタ109、およびミラー・テール・ポインタ107を保持していることに留意されたい。

【0038】本発明の一実施形態では、DMAC 38のムーバ64内にあるコマンド・バッファ・ベース・レジスタ(ベース・レジスタ)76が、主メモリ16内にあるコマンド・バッファ92のベース91のアドレス指定を可能にする。図4は、このベース・レジスタ76を例示した図である。これは、ビット0～12が0であり(すなわち8K境界)、ビット13～29が主メモリ16内の個々のページをアドレス指定する30ビット・レジスタである。

【0039】テール・ポインタ77およびヘッド・ポインタ79(DMAC 38のテール・レジスタ78およびヘッド・レジスタ80にそれぞれ保持されている)が、それぞれ、コマンド・バッファ92内に保持されているDMAコマンド・シーケンスのテール98およびヘッド100のアドレス指定を可能にする。DMAコマンド・シーケンスのヘッドは、このシーケンスの終端に関係しており、ここでDMAコマンドがコマンド・バッファに追加(格納)される。DMAコマンド・シーケンスのテールは、このシーケンスの終端に関係しており、ここでDMAコマンドがコマンド・バッファから取り出される(読み取られる)。以下に記載するように、コマンド・バッファはリング・バッファとしても構成される(すなわち、モジュロ・アドレス指定によりアドレス指定される)。

【0040】ベース・レジスタ76は、コマンド・バッファ92内のDMAコマンドを見つけるためにDMAC 38が発行するSBus DVMAアドレスを含む。

【0041】テール・レジスタ78内のテール・ポインタ77は、現在処理されているDMAコマンド(たとえばDMAコマンド102)のインデックスを提供する。テール・レジスタ78のテール・ポインタ77におけるそれぞれの増分は、主メモリにおける16バイト増分を表し、これは1つのDMAコマンドに対応する。テール・ポインタ値がゼロの場合は、そのコマンドがベース・

レジスタ76の指示するアドレスにあることを表す。

【0042】図5は、テール・レジスタ78を例示する図である。ビット0～3(すなわち16バイト境界)がゼロであり、ビット4～12がコマンド・バッファ92内のインデックス値を示すことに留意されたい。

【0043】ヘッド・レジスタ80は最後のDMAコマンド(たとえばDMAコマンド104)のインデックスを含み、これは主メモリ16内のコマンド・バッファ92で有効である。ヘッド・レジスタ80内のヘッド・ポインタによって与えられるアドレス指定は、テール・レジスタ78のテール・ポインタ77について説明したとおりである。ヘッド・レジスタ80を例示した図6を見るとわかるように、これは図5に例示したテール・レジスタ78と同じ形式である。

【0044】ヘッド・ポインタ79およびテール・ポインタ77の制御は、コマンド・バッファ92をリング・バッファとして構成する(すなわちモジュロ・アドレス指定が提供されている)ためのようなものであり、これによってポインタは、コマンド・バッファ92の一方の端から他方の端へ折り返す、すなわち、アドレスは0～n(ここでnは最大インデックス値に対応する)まで増加し、その後0に戻る。もちろんこのインデックス表示は、nから0という逆方向に進み、その後nに戻っても同じである。

【0045】DMAコントローラがアイドル状態であり、現在のコマンドについて進行中のすべてのDMAを完了した状況で、ヘッド・レジスタ80およびテール・レジスタ78にあるヘッド・ポインタ79およびテール・ポインタ77が、それぞれコマンド・バッファ92内の同じ位置を指示している場合、このコマンド・バッファには有効なDMAコマンドがなく、DMAコントローラはその時点で転送する必要のあるDMAコマンドがないことを理解する。

【0046】以下に説明するように、ヘッド・レジスタ80およびテール・レジスタ78が使用される方法は異なる。具体的には、ヘッド・レジスタ80はプロセッサ10によって管理されるが、テール・レジスタ78はDMAC 38によって管理される。

【0047】次に、図1に示したシステムの動作について、具体的にはDMAC 38の動作について、一般的に説明する。

【0048】前述のように、DMAC 38はローダ62およびムーバ64を備え、プロセッサ12、14がDMAC 38に対して提供するDMAコマンドに応答して動作することができる。DMAコマンドは、アラインメントに関する制約なしに、DVMAおよび入出力バス・アクセスに対して任意の開始アドレスを指定し、所与の最大値までの任意の転送長さを指定することができる。ムーバ64は、実行可能な最大幅の転送および最長のバーストを使用して、実行可能な最高の転送を処理す

る、読み取り値が好都合なアドレスで終わらない場合、実行可能な最速のバースト転送を使用するために、ムーバ64によって、DMAコマンドに指定されたデータよりも多くのデータが転送のどちらかの端で読み取られている可能性がある。過剰読み取りになると、次の適切な境界（たとえばSBusでは64バイト境界、入出力バスでは8バイト境界）まで何も読み取らない。書き込みは正確に処理される。したがって、ムーバ64は、予測されるデータを可能な限り最速の方法で書き込むに適した、小規模なSBusサイクルと入出力バス・サイクルの組合せであれば何でも使用する。

【0049】必要であれば、ローダ62は、完了状況の主メモリ内にあるDMAコマンドを更新することができる。DMAコマンドに指定されれば、ムーバ64がSBus割込みを生成することもできる。この割込みがプロセッサ10に達したときに、DMAコマンドは完了しており、メモリへの書き込みが要求されていることが保証されているが、主メモリに達する完了状況とプロセッサに達する割込みとの間のタイミングの関係は決められていない。

【0050】DMAの実行中、プロセッサは、主メモリ16内にあるコマンド・バッファ92の待ち行列にDMAコマンドを入れることが可能であって、單一プロセッサ入出力書き込みを使用して、すなわち新しいヘッド・ポインタをヘッド・レジスタ80に書き込むことによって、その存在についての信号をローダ62に送信することができる。ムーバ64が1つのコマンドに対するDMAを完了したときに、ヘッド・ポインタ79とテール・ポインタ77との間に何らかの差異があればそれを分析することによって、ローダは同時に次のDMAコマンドをムーバ・レジスタ74にロードし、プロセッサによる何らかの支援を待たずに、新たにDMAを開始することができる。言い換えれば、ローダ62は、ヘッド・ポインタ79とテール・ポインタ77との間の差異を使用して、DMAコマンドがコマンド・バッファ92からダウンロードされるのを待っていることを示す。

【0051】図7Aおよび図7Bは、プロセッサ10およびDMAC38の動作を例示した流れ図である。

【0052】図7Aは、プロセッサ10が実行する各ステップを表す。

【0053】ステップS1で、プロセッサは新しいDMAコマンド（たとえばDMAコマンド103）をコマンド・バッファ92に書き込む。プロセッサは、ミラー・ヘッド・レジスタ110の内容（すなわちミラー・ヘッド・ポインタ109）によって識別されるオフセットで、DMAコマンドをコマンド・バッファ92に書き込む。コマンド・バッファのベースは、ミラー・ベース・レジスタ106の内容（すなわちミラー・ベース・ポインタ105）によって識別される。

【0054】ステップS2で、プロセッサはミラー・ヘ

ッド・レジスタ110内のミラー・ヘッド・ポインタ109を増分する。コマンド・バッファ92内の次の位置は、ミラー・ヘッド・レジスタ110の内容を16バイトだけ増分することによって識別されるので、現時点で最後のDMAコマンド104を超えて次の位置を指示する。

【0055】ステップS3で、プロセッサはミラー・ヘッド・レジスタ110の内容とミラー・テール・レジスタ108の内容とを比較する。

【0056】これらが同等であれば（本実施の形態の場合はこれらが等しければ）、コマンド・バッファ内にはそれ以上スペースがない可能性があることを意味する。これを調べるには、ミラー・テール・レジスタ108の内容とテール・レジスタ78の内容とを確実に対応させる必要がある。したがってステップS4で、プロセッサはテール・レジスタ78の現在の内容をミラー・テール・レジスタ108にコピーする。最後にミラー・テール・レジスタ108が更新されてから、DMAC38がテール・レジスタ78内のテール・ポインタ77を変更した場合は、ミラー・ヘッド・レジスタ110の内容とミラー・テール・レジスタ108の内容が同等ではないはずである。ステップS3およびS4によって形成されるループは、ステップS3のテストによって、ヘッド・レジスタ110の内容とテール・レジスタ108の内容が同等でないことが示されるまで繰り返される。

【0057】ステップS3のテストによって、ミラー・ヘッド・ポインタ109がミラー・テール・ポインタ107と同等でないことが示されると、プロセッサはステップS5で、コマンド・バッファ92に書き込まれるDMAコマンドが他にもあるかどうかを判定する。他にもある場合は、ステップS1に制御が戻される。他にない場合、プロセッサはステップS6で、DMACのヘッド・ポインタ79がミラー・ヘッド・ポインタ109に対応するように更新するために、ミラー・ヘッド・レジスタ110の内容をDMAC38のヘッド・レジスタ80にコピーする。

【0058】その結果、プロセッサ10は、1つまたは複数のDMAコマンドをバッファ92のヘッドに追加し、DMACヘッド・レジスタ80への單一書き込みによって、DMAC38にすべてのコマンドを使用できることを示すことがわかる。これによって、入出力帯域幅要件を最小限に抑えることができる。

【0059】プロセッサ10は、コマンド・バッファ92内にスペースが存在するか否かを判定するために、各インスタンスでDMACテール・ポインタ77をミラーリングする必要はない。プロセッサ10はほとんどの場合に、前述のステップS1～S6を使用して、ミラー・テール・レジスタ108内に保持されているミラー・テール・ポインタ107を使用することができる。プロセッサ10は、ミラー・ヘッド・レジスタ110の内容と

ミラー・テール・レジスタ108の内容とを比較することで、コマンド・バッファ92にスペースがあるか否かを判定する。これらのミラー・レジスタ110および108内にあるミラー・ポインタ値が同等でない場合（この例では等しくない場合）、ミラー・ヘッド・レジスタ110のミラー・ヘッド・ポインタが指すアドレスにあるコマンド・バッファ92にはスペースがある。プロセッサは、DMAC 38のヘッド・レジスタ80に書き込む際にミラー・ヘッド・ポインタ109それ自体を更新するが、通常はミラー・テール・ポインタ107は修正しない。ミラー・ヘッド・ポインタ109とミラー・テール・ポインタ107との所定の関係が検出されると（この例では、ミラー・ヘッド・ポインタ値とミラー・テール・ポインタ値が等しい場合）、プロセッサ10は、テール・ポインタ77を最後にチェックしてからDMAC 38がこれを更新したか否かをチェックする。プロセッサ10は、テール・ポインタ77の現在値を見つけて、これをミラー・テール・レジスタ108にコピーするかまたは書き込むために、DMAC 38にあるテール・レジスタ78に対して单一読み取りアクセスを実行することができる。プロセッサ10がDMA要求を生成するのとほぼ同じ速度でDMAC 38が要求を処理していれば、プロセッサ10は、リング・バッファ92のローテーションが1回完了するごとにDMAテール・レジスタ78にアクセスするだけでよい。これによって、入出力帯域幅要件がさらに最小限に抑えられる。

【0060】図7Bは、DMAC 38が実行するステップを表す。

【0061】ステップS11でローダ62は、ヘッド・ポインタ79とテール・ポインタ77とを比較することができる。ヘッド・ポインタ79とテール・ポインタ77が同等であれば、現在、主メモリからムーバ64への転送を待っているDMAコマンドはないことが判定される。したがって、DMACローダ62は、ヘッド・ポインタ79とテール・ポインタ77との間に差異が生じるまで待機する。ここで同等であると言うことは、ポインタが、コマンド・バッファ内にある共通のDMAコマンド格納位置を効果的に指し示しているという意味であることがわかる。本実施形態の場合、これは両者が同じ値を有していることを意味する。ただし他の実施形態では、使用されるアドレス指定モードに応じて、互いに所定の関係を保った値を有することができる。

【0062】ヘッド・ポインタ79とテール・ポインタ77が同等でない場合（すなわち本実施形態では異なる値を有する場合）、ローダ62はステップS12でテール・ポインタ77を使用し、DMAの読み取り操作によって、コマンド・バッファ92内にある最後のコマンドにアクセスする。したがってDMACローダ62は、次のDMAコマンド（たとえばDMAコマンド102）を主メモリ16からムーバ・レジスタ74に転送することができる。

【0063】ムーバがDMAコマンドに指定されたデータの移動を完了するステップS13に続いて、ローダはステップS14で、「実行済み」または「完了」という指示を、DMAコマンドがそこから読み取られたばかりのコマンド・バッファ位置に書き込むことができる。プロセッサはこれを用いて、DMAC 38によってどのDMAデータが首尾良く転送されたか否かを監視することができる。

【0064】ステップS15で、DMACローダ62は、テール・ポインタ77がコマンド・バッファ92内の新しいテールを指し示すように更新する。

【0065】上記ステップS11～S15は、図1のローダ・コントローラ84が提供する機能を決めるものである。

【0066】前述のように、ヘッド・ポインタ79およびテール・ポインタ77は、モジュロ・アドレス指定を使用して、コマンド・バッファ92をリング・バッファとして構成することができる。この方式では、主メモリからDMAC 38への転送を待っているコマンドの文字列は、コマンド・バッファ用に割り振られたメモリ・スペース内で折り返すことができる。したがって、ヘッド・ポインタ79およびテール・ポインタ77の修正は、コマンド・バッファ92に割り振られたメモリ・スペースに基づく「モジュロ」方式で実行されるのが効果的である。

【0067】前述のように、プロセッサ10は、1つまたは複数のDMAコマンドのシーケンスをコマンド・バッファ92に書き込むのに使用することができる。このコマンド・シーケンスは、転送されたDMAコマンドの最後のアドレスだけをDMAC 38のヘッド・レジスタ80に送信して書き込むことができる。テール・ポインタ77は、次のコマンドのアドレスを形成するために使用されるので、DMAコマンドごとに更新される。DMACローダ62は、一連のDMAコマンドをコマンド・バッファ92からDMACムーバ64に転送するのに使用することができる。この場合、DMAコマンドのシーケンスがコマンド・バッファ92からDMACムーバ64に転送され、最後のDMAコマンドが完了した後に、割込みを生成するか、または主メモリ内の最終コマンドの完了状況を更新することができる。

【0068】次に、DMACローダ62およびDMACムーバ64で提供されるその他のレジスタについて簡単に説明する。

【0069】DMACローダ62内のその他レジスタ82は、現在のローダ状況を示すローダ状況レジスタを含むことができる。ローダは2つの主要な状態、すなわち受動状態と能動状態の間で移動する。パワーオン・リセット・コマンド、同期化リセット・コマンド、任意のデータ転送エラー・コマンド、あるいは直接プロセッサ・

コマンドによって、ローダを能動状態に移動させることができる。

【0070】能動状態では、ローダはDMAコマンドを処理中であっても処理中でなくともよい。すべてのコマンドが完了すると、ヘッド・レジスタ80の内容（すなわちヘッド・ポインタ79）はテール・レジスタ78の内容（すなわちテール・ポインタ77）と同等になる。ローダは、プロセッサ10が新しい値をヘッド・レジスタ80に書き込むのを能動状態で無期限に待機し、書き込まれた後、ローダは新しいDMAコマンドを処理する。能動状態は、ローダにとって通常の状態である。

【0071】DMACムーバ64内にあるムーバ・レジスタ74は、いくつかの異なるレジスタを備える。

【0072】DMAC_ACTI0Nレジスタは、現在のムーバ動作を決める。このレジスタは、DMACローダ62によって自動的にロードされ、主メモリ内にあるDMACコマンドのアクション・バイトに依存している。DMAC_ACTI0Nレジスタは、コマンドおよび状況情報を含む。これは、方向インジケータ、妥当性インジケータ、割込みイネーブル・インジケータ、およびライトパック・イネーブル・インジケータを含むことができる。方向インジケータは転送方向を示す。妥当性インジケータは、コマンドの妥当性をチェックすることができる。割込みイネーブル・インジケータは、DMA動作完了時にDMAムーバが割込みを生成するか否かを示す。ライトパック・イネーブル・インジケータは、コマンド完了後に、完了状況がコマンド・バッファに書き戻されるか否かを示す。DMACムーバはアクション・コマンドを完了すると、終了状況でDMAC_ACTI0Nを更新する。書戻し（ライトパック）が許可されると、DMACローダはDMAC_ACTI0Nレジスタ内で、更新されたDMAC_ACTI0Nを主メモリ16に書き戻す。DMAC_ACTI0Nレジスタは、その他の状況インジケータに使用する状況ビットをさらに含むことができる。

【0073】DMAC_DVMAレジスタは、DMAに使用されるSBus DVMAアドレスを決める。このDVMAアドレスは、システム入出力コントローラ30内にある入出力メモリ管理ユニットによって、実アドレスに変換される。このレジスタは、DMACローダ62によって、コマンド・バッファ92にあるDMAコマンドから自動的にロードされる。

【0074】DMAC_I0ADレジスタは、DMAに使用する入出力バスDVMAアドレスを決める。このレジスタも、ローダによって、コマンド・バッファ92にあるDMAコマンドから自動的にロードされる。

【0075】DMAC_C0UNTレジスタは、DMA C転送バイトをカウントする。これは、DMAコマンドに関して転送されるバイト数を与える。このレジスタも、DMACローダ62によって、コマンド・バッファ

92にあるDMAコマンドから自動的にロードされる。

【0076】前述のように、プロセッサ10は、ミラー・ベース・ポインタ105、ミラー・テール・ポインタ107、およびミラー・ヘッド・ポインタ109を、ミラー・ベース・レジスタ106、ミラー・テール・レジスタ108、およびミラー・ヘッド・レジスタ110内でそれぞれ維持する。図3に示すように、これらのレジスタは主メモリ内に保持することができる。別法として、これらをCPUレジスタで形成することができる。

【0077】図8は、DMAC 36のその他の態様をさらに詳細に例示した図である。具体的に言えば、図8はDMA転送機構85をより詳細に例示した図である。

【0078】図8に示すように、第1のSBusインターフェース202は、SBus 32からデータを受け取るために接続されている。第2のSBusインターフェース204は、SBus 32にデータを転送するために接続されている。これらのSBusインターフェースは、ムーバ・コントローラ86が供給するクロック信号CSによってクロックされている。図3に示すように、SBusは单一バス32である。ただし、SBusは制御ライン、アドレス・ライン、およびデータ・ラインを備えていてもよく、SBusインターフェース202および204は多重クロックト・クロック動作バッファとして構成することが可能であって、それぞれのバッファが1つまたは複数の制御ライン、アドレス・ライン、およびデータ・ラインに対して責任を負っている。SBusインターフェース202の出力は、マルチブレクサ210への第1の入力を形成するために、ライン206を介して供給される。このマルチブレクサ210への第2の入力は、アキュムレータ240の出力に接続されたバス208によって形成される。

【0079】マルチブレクサ210への第1の入力と第2の入力の選択は、ムーバ・コントローラ86による多重信号MS出力によって実行される。マルチブレクサ210の出力211は、バッファ・ステージ212に供給される。バッファ・ステージ212は、クロックト入力ゲート214、バッファ記憶域（たとえば容量が128×8バイトのランダム・アクセス・メモリ）、およびクロックト出力ゲート218を含む。クロックト入力ゲート214およびクロックト出力ゲート218は、ムーバ・コントローラ86からのクロック信号CSを受け取る。

【0080】バッファ・ステージ212の出力は、第2のマルチブレクサ230への第1の入力226と、保持レジスタ222への入力224となる。保持レジスタ222は、ムーバ・コントローラ86からのクロック信号CSによってクロックされる。保持レジスタ222の出力は、第2のSBusインターフェース204への入力220を形成する。

【0081】第2のマルチブレクサ230への第2の入

力228は、第1の入出力バス・インターフェース250からの出力によって形成される。

【0082】第2のマルチブレクサ230からの第1の出力と第2の出力の選択も、ムーバ・コントローラ86からの多重信号(MS)によって実行される。第2のマルチブレクサ230の出力232は、シフタ/ローテータ234に供給される。このシフタ/ローテータ234は、選択可能なパレルシフトを行うための制御可能なパレル・シフタである。シフタ/ローテータ234は、ムーバ・コントローラ86からシフト制御信号236を受け取る。シフタ/ローテータ234からの出力238は、アキュムレータ240に供給される。アキュムレータ240は、ムーバ・コントローラ86からバイト・イネーブル信号242を受け取り、ムーバ・コントローラ86からフェーズ信号244も受け取る。アキュムレータ240からの出力は、第1のマルチブレクサ210への第2の出力208を形成し、第2の入出力バス・インターフェース248への入力246も形成する。第1の入出力バス・インターフェース250と第2の入出力バス・インターフェース248も、ムーバ・コントローラ86からのクロック信号CSによってクロックされる。入出力バス・インターフェース248からの出力252は、入出力バス40にデータを置くことであり、第1の入出力バス・インターフェース250への入力254は、入出力バス40からのデータを受け取ることである。

【0083】次に、ムーバ転送機構85の動作について下記に記載する。ムーバ・コントローラ86は、ムーバ転送機構85の動作を制御するために、ムーバ64のムーバ・レジスタ74内にあるデータに応答して動作することに留意されたい。

【0084】図9は、シフタ/ローテータ234を示す概略図である。図4に示すように、シフタ/ローテータ234への入力232およびシフタ/ローテータ234からの出力238は、マルチバイト出力である。ムーバ・コントローラ86からのシフト入力236は、シフト制御信号236によって選択されたシフタ/ローテータ234内のパレルシフト動作を制御するために使用することができる。

【0085】図10は、アキュムレータ240を例示した図である。シフタ/ローテータ234からアキュムレータ240への入力238は、平行にバイト・マスク239へ供給される。バイト・マスクを使用して、バイト・イネーブル信号242にしたがって、その入力から1つまたは複数のバイトを選択することができる。アキュムレータ・レジスタ241を使用して、フェーズ信号244により、バイト・マスク239が供給するバイトを選択的に累積し、出力することができる。この例では、マスキングはバイトごとに実行されるが、他の実施形態では、例えばビットごとに選択することができるに留意されたい。フェーズ信号はアキュムレータ・レジス

タ241の異なるフェーズの動作を累積し、その後累積されたビットを出力するように制御する。バイト・イネーブル信号242およびフェーズ信号244は、ムーバ・コントローラ86によって供給される。アキュムレータ・レジスタ241からの出力は、ライン208および246に供給される。

【0086】図8では、斜線で印を付けたように、すべての内部データ・バス・ラインが64ビット幅(すなはち8バイト幅)である。ただし、バス32および40上のデータ・ラインおよびアドレス・ラインは、別のバス・インターフェースを使用して別々に制御できることが可能であって、この場合、バスとインターフェースとを結ぶラインが内部データ幅とは異なる場合があることに留意されたい。さらにまた、この例の内部データ幅は、可能な内部データ幅の単なる一例であって、代わりに他の内部データ幅(たとえば32ビットまたは128ビット)を使用してもよいことを理解されたい。

【0087】図11は、ソース・バス(S Bus 32または入出力バス40のいずれでもよい)とそれらのバスの他方にある宛先との間で実行可能な転送を例示した図である。この例では、実施形態が64ビット幅のデータ・バスであるので、8バイト幅のデータ・ソースが表示されている。ただし、上記で提案したように、他の実施形態では他のデータ幅が採用されることがある。もちろんDMA転送は、DMAC 38の位置に応じて、他のバス間でも実行可能である。

【0088】ソース情報260はワード0のバイト4から開始していることに留意されたい。これに対して宛先情報は、ワード10のバイト6から開始している。したがって、単にバイトをワードごとにソースから宛先に転送することはできないことがわかる。ムーバ転送機構は、ソース・アドレスから宛先アドレスへのデータの転送を非常に効率よく実行するように構成される。

【0089】次に、シフタ/ローテータ234とアキュムレータ240の動作について、図12を参照しながら説明する。

【0090】図12で、第1列は、転送の第1期、第2期、第3期、および第4期でのシフタ/ローテータ234への入力232を表す。第2列は、これら4期におけるシフタ/ローテータ234の出力238を示す。第3列は、第1期～第4期について、第1フェーズおよび第2フェーズでアキュムレータ240に供給されるバイト・イネーブル信号242を表す。第4列は、4期の各フェーズにおける、アキュムレータ・レジスタ241の内容を例示する。第5列の矢印は、第1期～第4期それぞれの第1フェーズ中に、アキュムレータ240の内容を245で出力することを表す。第1列に示した1, 1, 1, 2, 2, 1などの参照番号は、第1期の第1フェーズ、第1期の第2フェーズ、第2期の第1フェーズなどを表す。次に、図12についてより詳細に説明する。

【0091】以上のことから、第1期(1. 1で示す)では、シフタ・ローテータ234への入力はソース・ワード0の内容(すなわち***ABCDE)である、シフタ・ローテータ234はこの情報を右へ2バイト分バーレルシフトして、238の列1. 1に示した信号(すなわちDE***ABC)を提供する。この情報は、シフタ・ローテータ234の出力で即時に使用できる。同時に、この第1フェーズでは、242の行1. 1に図示されたバイト・イネーブル信号(00000111)が、アキュムレータ240のバイト・セレクタ239に供給される。これによってアキュムレータ240に、アキュムレータ・レジスタ241内に、バイトイネーブル信号の「1」で識別される位置で使用可能なバイトを累積させ、バイトイネーブル信号の「0」に対応するバイト位置にある内容は選択しないようにさせる。アキュムレータ・レジスタ241の内容は、第1の宛先バイト10(すなわち***ABC)を形成するために(245の行1. 1に示すように)即時に出力される。

【0092】1回の実行で3バイトが出力されるように表示されている。これを望まないあるいはこれが不可能な他の実施形態の場合、ムーバ・コントローラ86は信号を供給して、出力に3バイトが1回に1バイト選択されるようにすることができる。

【0093】行1. 2に示すように、動作の第2フェーズでは、シフタ・ローテータ234への入力およびここからの出力はそのままである。ただし、アキュムレータに供給されるバイト・イネーブル信号242は11000000に変わる。その結果、アキュムレータ240に入力される最初の2バイト(DおよびE)がセレクタ239によって選択され、アキュムレータ・レジスタ241に累積される。ただし、これらの信号は出力されず、代わりに動作の次のフェーズになるまでアキュムレータ・レジスタ内に保持される。

【0094】動作の次のフェーズは、ステップ2の第1フェーズを形成する。行2. 1に示すように、シフタ・ローテータ234への入力232は、ソース・バスからのワード1(すなわちFGHIJKL)に関する情報を変わる。シフタ・ローテータ234からの出力238は、この入力を右へ2バイト分バーレルシフトしたもの(すなわちLMFGHIJK)に対応する。このフェーズで、図7の242に示すように、バイトイネーブル信号が00111111に変わり、これによってセレクタ239は、アキュムレータ・ユニット241で累積するために信号FGHIJKを選択する。これで宛先への第2のワードの供給は完了し、次いで列2. 1の245に矢印で示されたように、8バイトのワードとして出力される。

【0095】図12での考察により、このプロセスがどのようにして2. 2. 3. 1. 3. 2. および4. 1の各フェーズへと続行されるかが理解されよう。第4の宛

先バイト(バイト13)に出力される情報は、フェーズ4. 1が終わった時点ですでにアキュムレータに格納されているので、フェーズ4. 2では必要ないことに留意されたい。したがってこの例では、列4. 1の245に矢印で示されているように、この情報は即時に出力される。他の例で、1ワードが8バイトより少ないワードが出力される場合、特定の実施形態により、1バイト、2バイト、および/または4バイトのグループで出力することができる。

【0096】したがって、バス32とバス40との間での転送、またはバス40とバス42との間での転送の場合、ソースアラインメントから宛先アラインメントに転送するために、情報がいかに効率よくバーレルシフトされて累積されるかが理解されよう。

【0097】バイトの詳細な動きは、実施の形態によつて異なる。たとえば、どちらのバスもバースト・モードで動作している場合、前述のようにアキュムレータ240を2つのフェーズで動作させるようになることができる。第1フェーズでは、シフタから入出力データ・ワードxのnバイトを入手する。第2フェーズでは、入出力データ・ワード(x+1)から(8-n)バイトを入手する。次いでアキュムレータの内容がバッファ216に転送され、(x+1)からnバイトを即時に入手する、といった具合である。値nはシフトに関係し、ソース・バスおよび宛先バス上にある開始アドレスのバイト・オフセットに依存する。

【0098】シフト信号236、バイト・イネーブル信号242、およびフェーズ信号244は、すべてムーバ・コントローラ86によって提供されることに留意されたい。このムーバ・コントローラ86を使用して、ソース・アドレスおよび宛先アドレス、これらの信号を生成するための転送方向、ならびに図8に例示した他のユニットに供給されるクロック信号CSおよび多重信号MSを含む、ムーバ・レジスタ74に格納されたデータに応答することができる。

【0099】図13は、ムーバ・コントローラの動作をまとめた構成図である。ステップS21では、ムーバ・コントローラがムーバ・レジスタ74からソース・アドレスおよび宛先アドレスを取り出す。S22では、バス32とバス40との間の転送方向を決定する。この情報は、一般にムーバ・レジスタ74のうちの1つの転送方向指示によって決定される。ただし代わりに、関係するバス・プロトコルを使用することができるソース・アドレスおよび宛先アドレスから直接取り出すことも可能である。

【0100】S23では、ムーバ・コントローラが、シフタ・ローテータ内で実行されるシフトおよび転送されるブロック・シーケンスのサイズを決定する。

【0101】この情報は、基礎のしっかりしたアルゴリズム、所与の正確なソース・アドレスおよび宛先アドレ

ス、ならびに転送方向により、容易に計算できることに留意されたい。転送用ブロックの決定は、バス間での転送をできる限り効率的に実行するように確定される。

【0102】図14に示すように、転送は異なるブロック・サイズで実行することが可能であって、使用可能なバス・プロトコルおよび使用可能なハードウェア設計幅の範囲内で実行される。これらのサイズによるブロック転送の可用性に依存する。これは自由に選択することができる。

【0103】図14に示すように、転送ブロックは1バイト・サイズ、8バイト・サイズ、および64バイト・サイズが可能であると考えられる。他の実施形態では、1バイトよりも量が少ない場合にビット転送も可能である。図14に概略を示したように、理論上の転送には、以下のステップが含まれる。

【0104】ステップS31では、「8バイト」ブロック境界に達するまで、1バイト・サイズ・ブロックでの初期転送が実行される。

【0105】ステップS32では、「64バイト」ブロック境界に達するまで、8バイト・ブロック・サイズでバイトを転送することができる。

【0106】ステップS33では、転送の終わりに近づき、8バイト・ブロック・サイズに戻る必要が生じるまで、64バイト・ブロックで転送が実行される。

【0107】ステップS34では、残りのバイトを1バイト・サイズのブロックで転送する必要が生じるまで、8バイト・ブロックが転送される。

【0108】ステップS35では、残りのバイトが転送される。

【0109】前述のように、ステップS31より前に最初のステップを設定することおよび/またはステップS35より後に最終ステップを設定することによって、ビットごとの転送を行うことができる。

【0110】以上のことから、実行される転送のサイズが決まっていてソース・アドレスと宛先アドレスが正確であり、8バイト境界および64バイト境界を基準に分布されていれば、上記の1つまたは複数ステージの組合せは、ムーバ・コントローラ86が容易に決定することができる。好適なアラインメントがすでに実行されている場合、図9に示した1つまたは複数のステップが省略できることは明らかである。

【0111】図13に戻ると、いったんステップS23が決定されると、ステップS24で制御信号の詳細なシーケンスを順番に生成し、ステップS25で転送を実行することができる。

【0112】前述のように、バス32とバス40との間の転送方向を決定する必要がある。SBus32から入出力バス40へ転送しても、その逆でもよい。

【0113】多重信号MSを第1のマルチブレクサ210および第2のマルチブレクサ230に提供することに

よって、転送方向の制御が実行される。図15は、入出力バス40からSBus32へのデータ転送を示す図である。これを実行するために、第1のマルチブレクサ210を制御して、アキュムレータ240の出力である第2の入力を選択する。第2のマルチブレクサ230を制御して、入出力バス・インターフェース250から第2の入力228を選択する。

【0114】その結果、入出力バス・インターフェース250で入出力バス40から受け取られた転送されるべきバイトが第2のマルチブレクサ230に供給され、今度はこれがそれらのバイトをシフタ/ローテータ234に供給する。図6および図7に開いてすでに述べたシフタ/ローテーションおよび累積プロセスがシフタ/ローテータおよびアキュムレータで実行され、アキュムレータから出力されたバイトが第1のマルチブレクサ210に供給される。SBusインターフェース240を介してSBus32へ転送するために保持レジスタ222へ転送されるデータ・ブロックを形成するために、第1のマルチブレクサ210からの出力がバッファ216でバッファリングされる。バッファ212および保持レジスタ222は、異なるバス速度およびバス・アクセスの可用性を考慮に入れるために、入出力バスとSBus32との間でのデータのバッファリングを可能にする。

【0115】図15では、ライン201、206、226、246、および252は破線で表示されていることに留意されたい。これは、データが入出力バス40からSBus32に転送される場合に、データがこれらのバスに沿って転送されないことを表している。

【0116】図16は、データ転送がSBus32と入出力バス40との間で実行される場合の、代替状況を例示した図である。この場合、ライン201、206、226、246、および252は実線で表示されており、これに対してライン254、228、208、224、220、および203は破線で表示されていることに留意されたい。これは、SBus32から入出力バス40へ情報が転送される際に、データは破線に沿って流れないことを示している。

【0117】したがって、図16に示された状況では、データはSBus32からライン201を通じてSBusインターフェース202で受け取られる。SBusインターフェース202からのデータは、ライン206を介して第1のマルチブレクサ210に供給される。この場合、多重信号MSは、第1の入力を選択するために第1のマルチブレクサ210に供給される。したがって、マルチブレクサ210の入力206で受け取られたデータはその出力211に供給され、バッファ・ユニット212でバッファリングされる。このデータは、シフタ/ローテータおよびアキュムレータ240を通じて入出力バス40へ転送される一定に流れるよう、バッファ・ユニット212から出力される。したがって、バッ

ファ・ユニット212から出力されるデータは、第2のマルチブレクサ230の第1の入力226に供給される。多重信号MSは、その第1の入力226を選択するために、第2のマルチブレクサ230に供給される。したがって、マルチブレクサ230の第1の入力226のデータは、その出力232に供給される。図6および図7に関する前述のシフト/ローテーションおよび累積動作は、シフタ/ローテータ234およびアキュムレータ240によって実行される。そこから出力されるデータは、ライン252を介して入出力バス40へ出力するために、ライン245および246を介して入出力バス・インターフェース248に供給される。

【0118】したがって、ソース・バスからデータを入手し、これを宛先バスに出力するための所望のアラインメントに再アラインメントするために、FIFOバッファ216と調和して使用されるシフト/累積機構234～240について記述してきた。具体的な例では、2つのバスはSBus-32と入出力バス40である。ソースはこれらバスのうちのいずれか一方、宛先はこれらバスのうちの他方とすることができます。ただし本発明は一般に、どのような好適なバス・プロトコルの下でも動作可能なバスに適用可能であることを理解されよう。

【0119】この例では、SBusで使用される本来のバイト・アラインメントは64バイトであり、この機構は可能であればどこでも64バイトのバースト転送を使用できるようにする。本実施例の入出力バスは8バイトにアラインメントされる。本実施例では、入出力バス転送は512バイト長さまで可能であり、8バイト境界上で開始および終了する。ただし他の実施例では、サイズの異なるバスの転送が実行できる。第1のモード(DMA読み取りモード)の場合、入出力バスがソースであり、SBusは宛先である。シフタ/ローテータ234は、開始ソース・アドレスおよび開始宛先アドレスの相対的バイト・アラインメントに応じてデータをローテーションする、8バイトのバレル・シフタである。入出力バス転送は、ソース・アドレスから読み取る際に使用することができます。図12を参照すると、転送の始めおよび終わりに何らかの特別なデータを読み取ることに留意されたい。バイトは、宛先アドレスが8バイトにアラインメントされるまで、アキュムレータ内で入手される。次いでこのアキュムレータの内容が、バッファ216に転送される。その後このバッファの内容は、64バイト・アドレス境界に達するまで、バイト転送または8バイト転送を使用してSBusに書き込まれるが、この時点で64バイトのバースト転送が使用される。

【0120】両方のバスがバースト・モードの場合、アキュムレータは2フェーズで動作する。第1フェーズでは、シフタから入出力データ・ワードxのnバイトを入手する。第2フェーズでは、入出力データ・ワード(x+1)から(8-n)バイトを入手する。次いでこのア

キュムレータの内容がFIFOバッファに転送され、同時に(x+1)からnバイトを入手する、といった具合である。値nはシフトに関係し、ソース・バスおよび宛先バス上にある開始アドレスのバイト・オフセットに依存する。

【0121】DMA書き込みモードでの動作は、SBusがソースであり入出力バスが宛先であること以外は同じである。SBus読み取りは必ず64バイト・バースト上で開始する。転送の始めと終わりに何らかの特別なバイトが読み取られることがある。データはFIFOバッファ216に読み込まれ、シフタ/ローテータ234およびアキュムレータ240を通って進み、入出力バスに書き込まれる。8バイト・アドレス境界に達するまで個々のバイト(またはバイトのグループ)が入出力バスに書き込まれ、この時点で残りが8バイト未満になるまで8バイトのバースト転送が使用される。その後残りのバイトは1バイトずつ転送される。

【0122】ここまで、2つのバス間でバイトを転送するための非常に効果的で柔軟な方法について説明してきた。バイト境界、8バイト境界、および64バイト境界での転送を参照しながら説明してきたが、転送はその他の境界でも実行可能であり、バイトのサブユニット、たとえばビットごとでも実行可能であることが理解されよう。

【0123】以上、本発明を含むコンピュータ・システムの特定の例示的構成について説明してきたが、本発明がこうした構成を有するコンピュータ・システムのみに適用可能なものではないことを理解されよう。

【0124】実際、本発明の特定の実施形態について説明してきたが、添付の特許請求の範囲に定義されているように、本発明の精神および範囲を逸脱しなければ、多くの修正、追加、および/または置換が実行できることを理解されよう。

【図面の簡単な説明】

【図1】本発明を組み込んだコンピュータ・システムの例示的構成を示す概略構成図である。

【図2】図1のシステムのDMAコントローラを示す概略構成図である。

【図3】主メモリにおけるコマンド・バッファを示す概略図である。

【図4】ベース・レジスタを示す図である。

【図5】テール・レジスタを示す図である。

【図6】ヘッド・レジスタを示す図である。

【図7】DMAコマンドをDMAコントローラに転送するための、図1のコンピュータ・システムの動作を図示した流れ図である。

【図8】図2のDMAコントローラのDMA転送機構を示す概略構成図である。

【図9】図8の機構のシフタ/ローテータを示す概略構成図である。

【図10】図8の機構のアキュムレータを示す概略図である。

【図11】データを転送するためのソース・バイトおよび宛先バイトのアラインメントを示す図である。

【図12】シフタ／ローテータおよびアキュムレータの動作を示す図である。

【図13】データを転送するための制御動作を示す流れ図である。

【図14】可能なブロック転送動作を示す図である。

【図15】第1バスと第2バスとの間で転送するための、図8の機構の構成図である。

【図16】第1バスと第2バスとの間で転送するための、図3の機構の構成図である。

【符号の説明】

1 0 コンピュータ・システム

1 2 第1プロセッサ

1 4 第2プロセッサ

1 6 主メモリ

1 8 システム・コントローラ

2 0 メモリ・インターフェース

2 2 第1バス

2 4 制御ライン

2 6 制御ライン

2 8 制御ライン

2 9 制御ライン

3 0 システム入出力コントローラ

3 2 システム・バス (S Bus)

3 3 周辺バス (P Bus)

3 4 スロット

3 6 プロセッサ入出力ブリッジ

3 8 ダイレクト・メモリ・アクセス・コントローラ (DMA C)

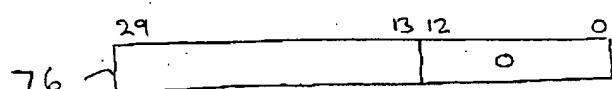
4 0 入出力バス

4 6 周辺装置

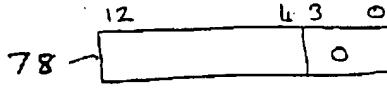
4 8 周辺装置

5 0 周辺装置

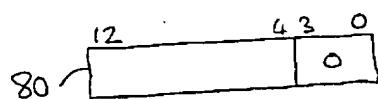
【図4】



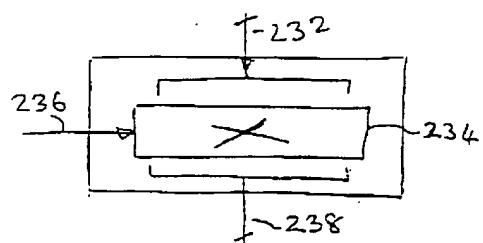
【図5】



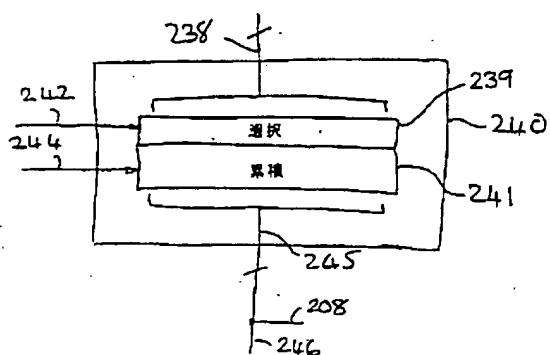
【図6】



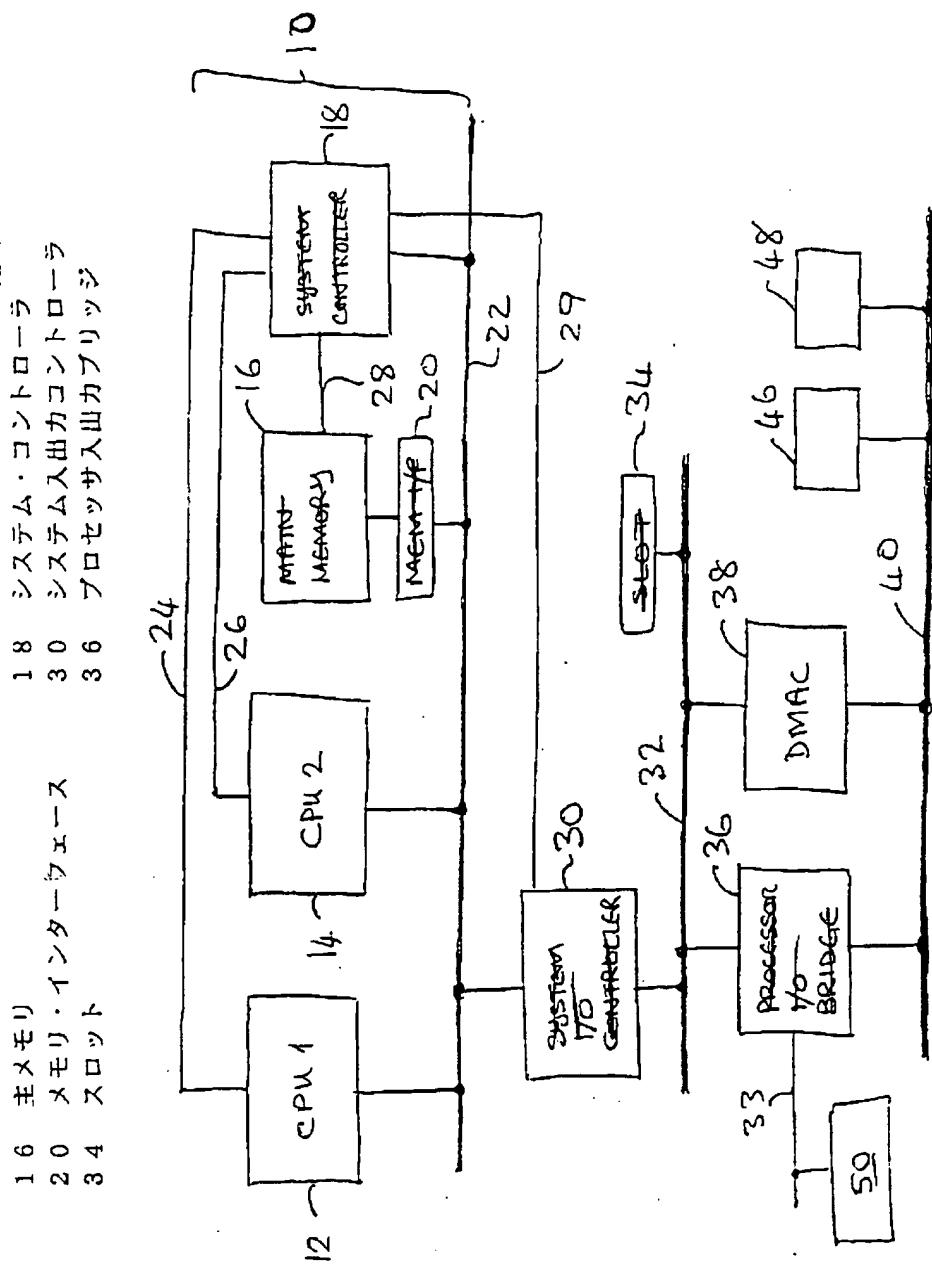
【図9】



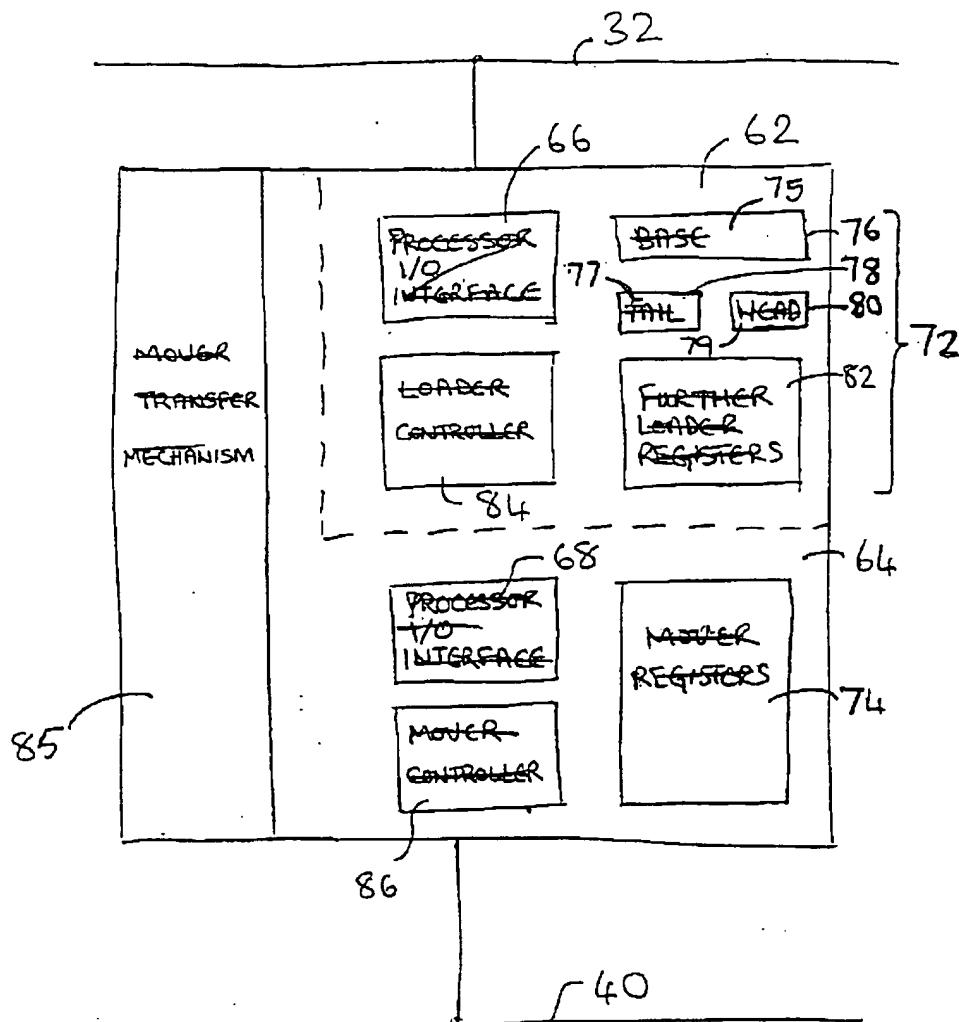
【図10】



【図1】



【図2】



66 プロセッサ入出力インターフェース

68 プロセッサ入出力インターフェース

74 ムーバ・レジスタ

75, 76 ベース

77, 78 テール

79, 80 ヘッド

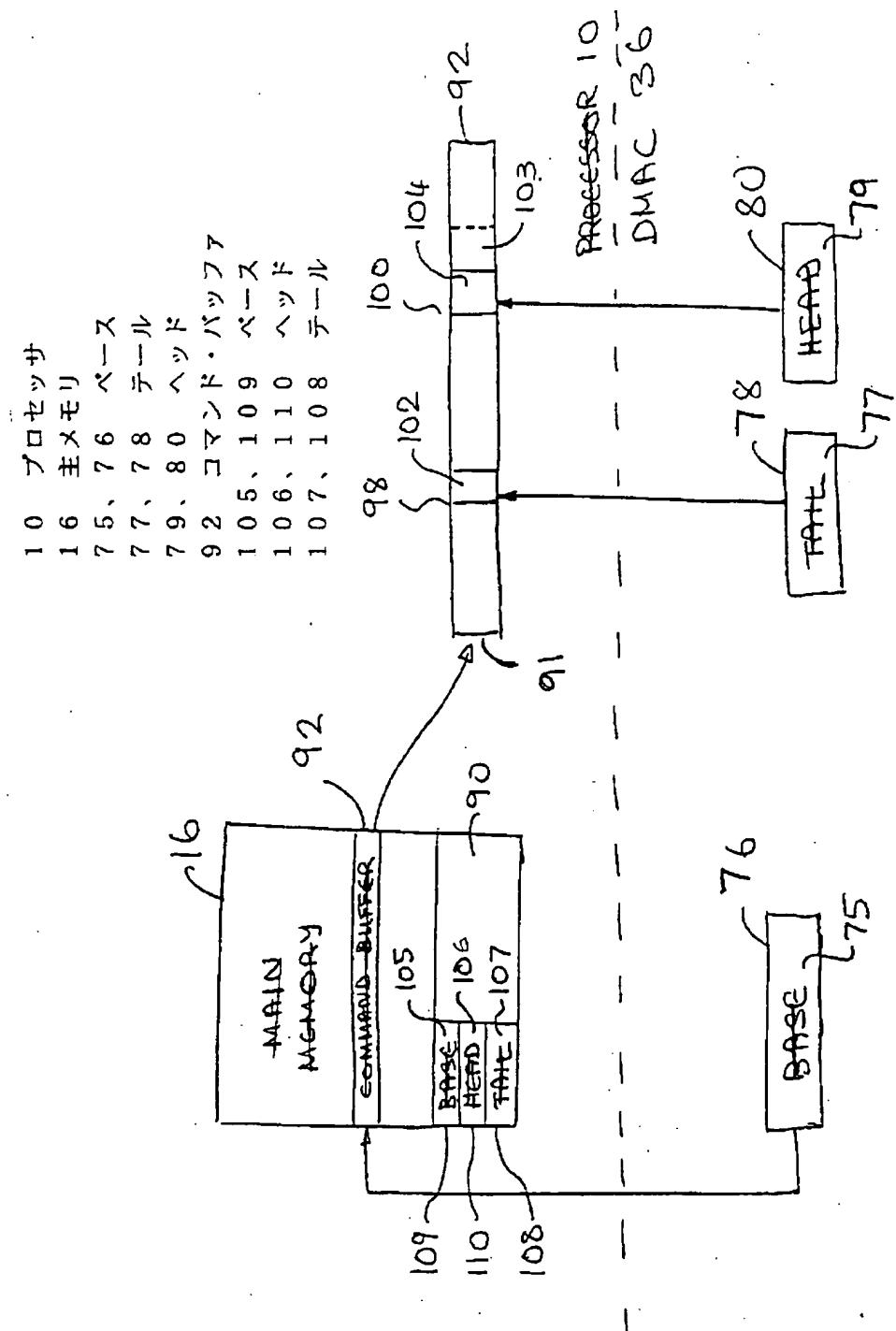
82 その他ローダ・レジスタ

84 ローダ・コントローラ

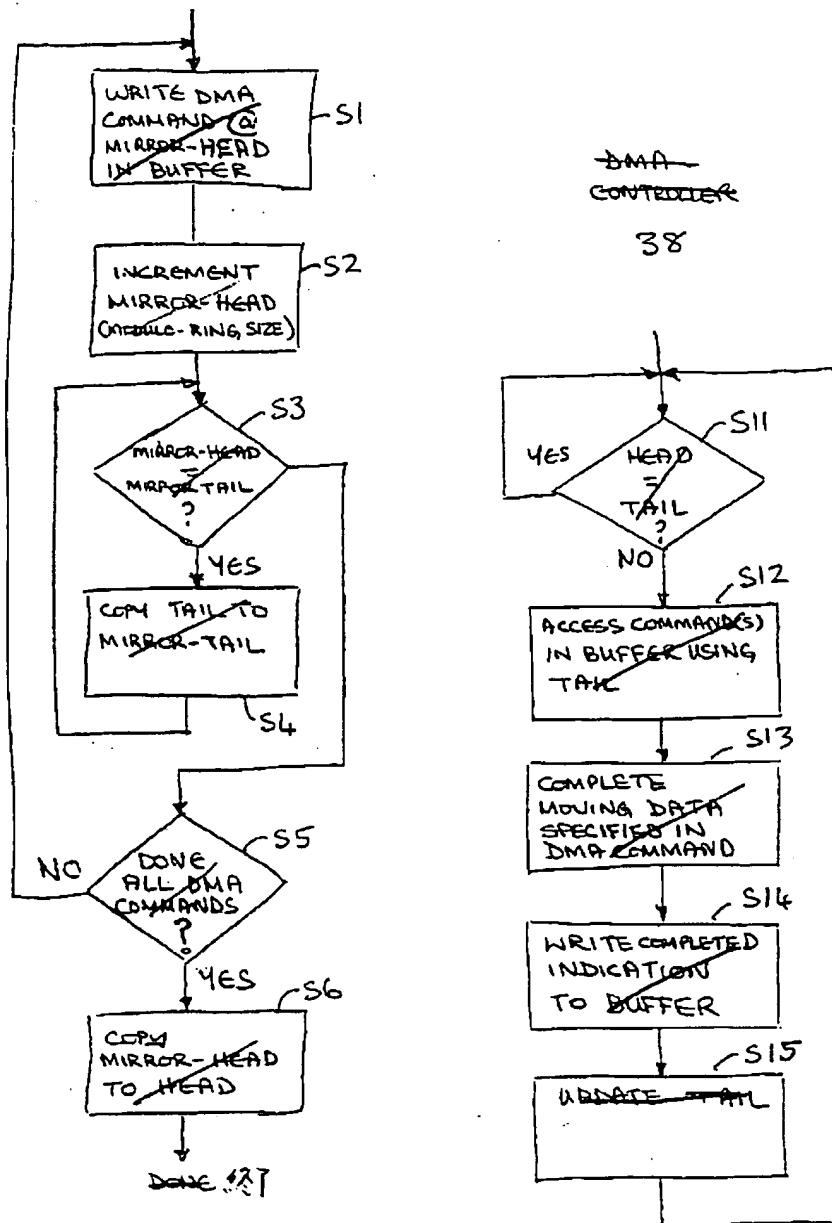
85 ムーバ転送機構

86 ムーバ・コントローラ

【図3】



[図7]



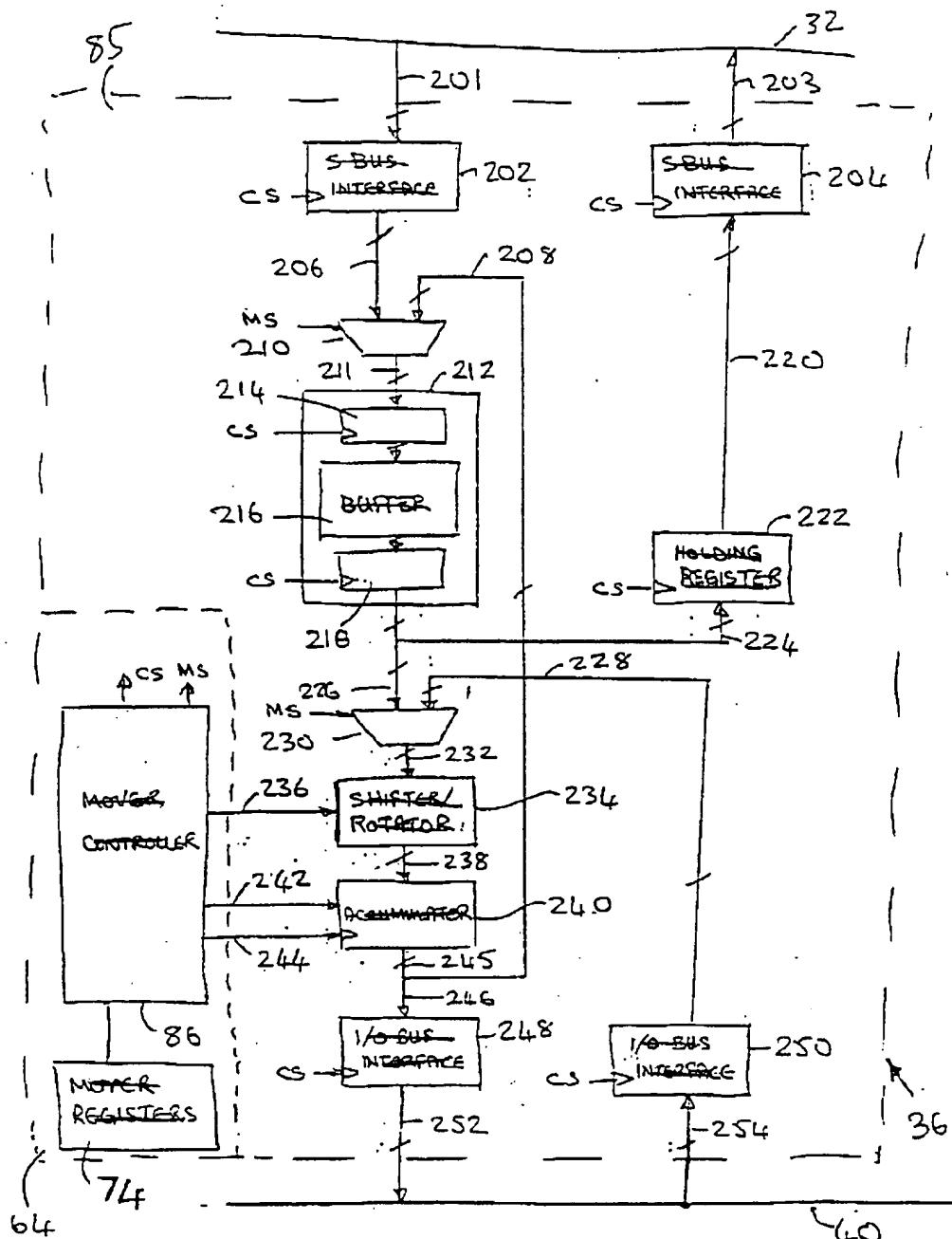
PROCESSOR 10

A

B

S1 DMAコマンドとミラー・ヘッドをバッファに書き込む
 S2 ミラー・ヘッドとミラー・ヘッドを増分 (モジュロ・リング・サイズ)
 S3 ミラー・ヘッドとミラー・テールが等しいか
 S4 テールをミラー・テールにコピー
 S5 すべてのDMAコマンドを実行したか
 S6 ミラー・ヘッドをヘッドにコピー
 S11 ヘッドとテールが等しいか
 S12 テールを使ってバッファ内のコマンドにアクセス
 S13 DMAコマンドに指定されたデータの移動を完了
 S14 完了指示をバッファに書き込む
 S15 テールを更新
 10 プロセッサ
 38 DMAコントローラ

【図8】



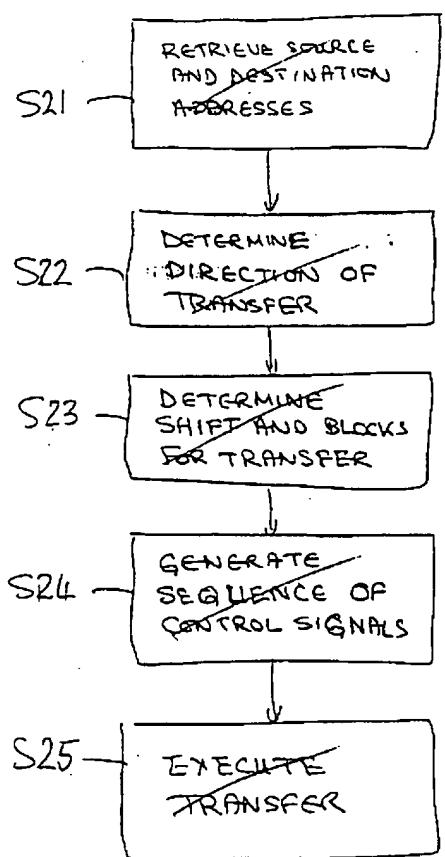
| | | | |
|-------|----------------|-------|----------------|
| 7 4 | ムーバ・レジスタ | 8 6 | ムーバ・コントローラ |
| 2 0 2 | S-Bus インターフェース | 2 0 4 | S-Bus インターフェース |
| 2 1 6 | バッファ | 2 2 2 | 保持レジスタ |
| 2 3 4 | シフタ/ロータ | 2 4 0 | アクチュエータ |
| 2 4 8 | 入出力バス・インターフェース | | |
| 2 5 0 | 入出力バス・インターフェース | | |

[図11]

【图12】

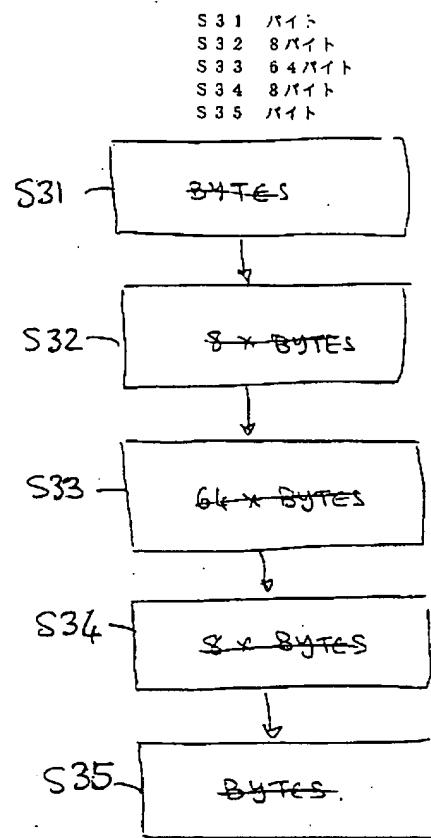
| | 232 ↓ | 238 ↓ | 242 ↓ | 241 ↓ | 245 ↓ |
|-----|---------------|-----------------|-----------|-----------------|----------|
| 1.1 | ABCDE | DEABC | 00000111 | 00000ABC | → |
| 1.2 | | | 11000000 | DE111111 | |
| 2.1 | EFGHIJKLM | LMFEGHIJKLM | 00111111 | DEFEGHIJKLM | → |
| 2.2 | | | 110000000 | LM111111 | |
| 3.1 | NOPQRSTUVWXYZ | TUNOPQRSTUVWXYZ | 001111111 | LMNOPQRSTUVWXYZ | → |
| 3.2 | | | 110000000 | TU111111 | |
| 4.1 | WVWXYZ | VWXYZZ | 001111110 | TUVWXYZ | → |

【図13】



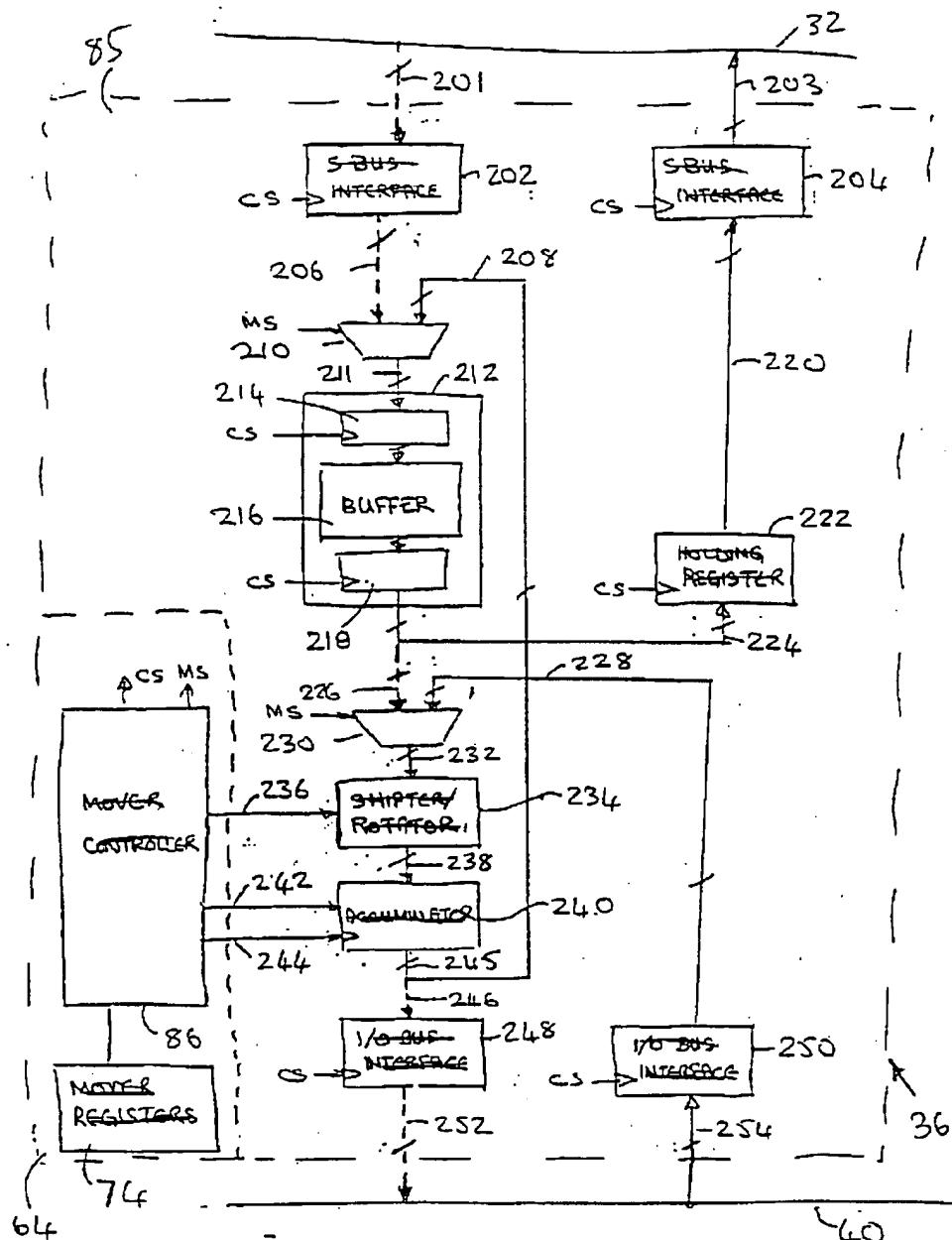
S21 ソースおよび宛先アドレスの取出し
 S22 転送方向を決定
 S23 転送のためのシフトおよびブロックを決定
 S24 例制信号シーケンスを生成
 S25 転送を実行

【図14】



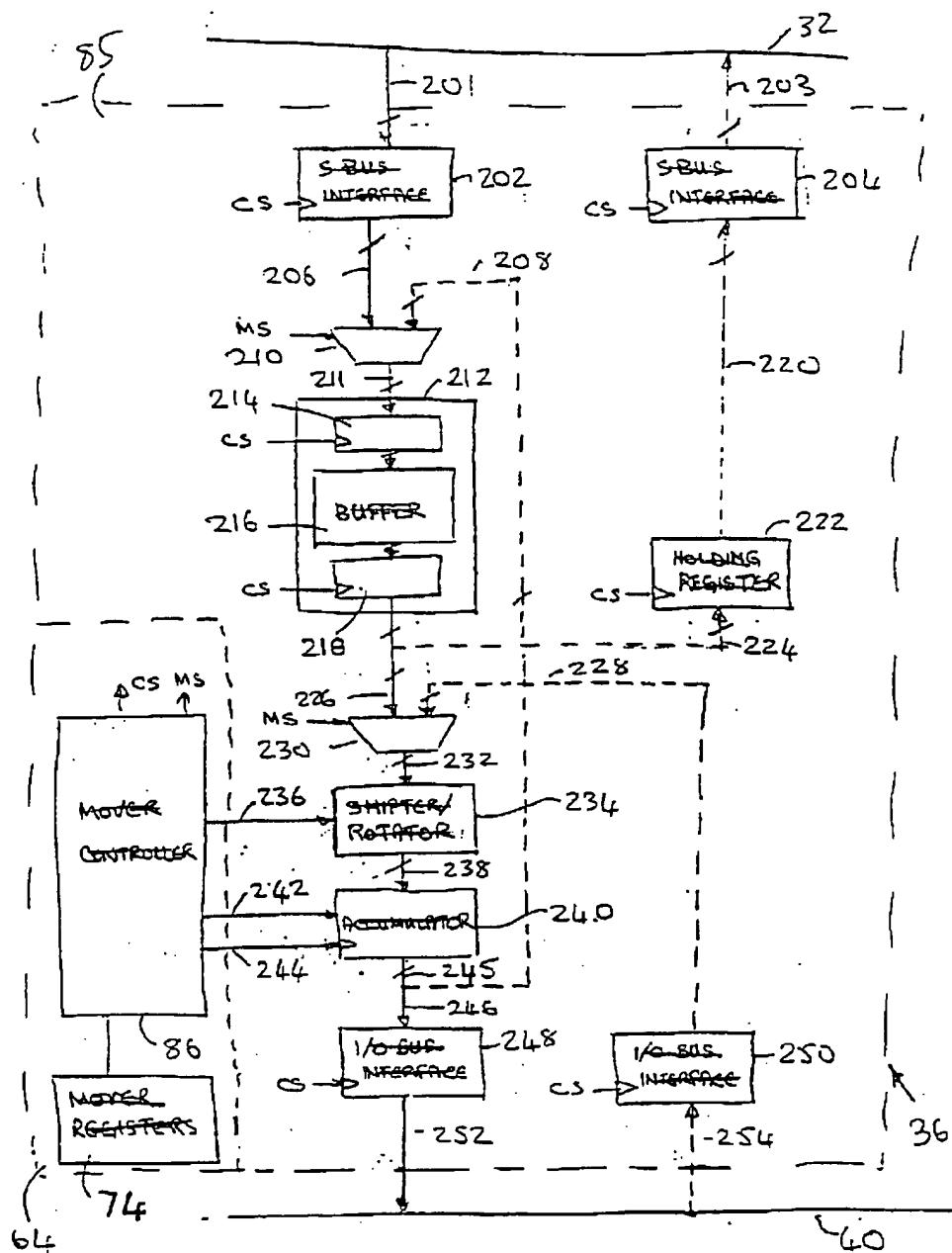
S31 バイト
 S32 8バイト
 S33 64バイト
 S34 8バイト
 S35 バイト

【図15】



| | | | |
|-----|----------------|-----|----------------|
| 74 | ムーバ・レジスタ | 86 | ムーバ・コントローラ |
| 202 | S-Bus インターフェース | 204 | S-Bus インターフェース |
| 216 | バッファ | 222 | 保持レジスタ |
| 234 | シフタ/ローター | 240 | アキュムレータ |
| 248 | 入出力バス・インターフェース | 250 | 入出力バス・インターフェース |
| 250 | 入出力バス・インターフェース | | |

〔图16〕



74 ムーバ・レジスター
 202 SBロスインターフェース
 216 バッファ
 234 シフト／ローテータ
 248 入出力バス・インターフェース
 250 入出力バス・インターフェース

86 ムーバ・コンピュータ
204 SBUSインターフェース
222 保持レジスター
240 アキュームレータ

フロントページの続き

(71)出願人 591064003

901 SAN ANTONIO ROAD
PALO ALTO, CA 94303, U.
S. A.

(72)発明者 アンドリュ・クロスランド

イギリス国・エイチビイ17 8イーエックス・バッキンガムシャー・ハーデンハム・シアーストック・118